



# Design of 1-tape 2-symbol reversible Turing machines based on reversible logic elements

Jia Lee<sup>a,\*</sup>, Rui-Long Yang<sup>a</sup>, Kenichi Morita<sup>b</sup>

<sup>a</sup> Chongqing Key Laboratory of Software Theory & Technology, College of Computer Science, Chongqing University, Chongqing, China

<sup>b</sup> Graduate School of Engineering, Hiroshima University, Higashi-Hiroshima, Japan

## ARTICLE INFO

### Article history:

Received 21 December 2011

Received in revised form 26 June 2012

Accepted 9 July 2012

Communicated by O. Watanabe

### Keywords:

Reversible Turing machine

Reversible logic element

Reversibility

Asynchronous operation

## ABSTRACT

This paper proposes a novel scheme for constructing reversible Turing machines (RTMs) via various reversible logic elements. A reversible logic element is similar to a conventional reversible logic gate, except that the element also carries a 1-bit memory that can store binary states. The usage of internal states allows much more efficient and straightforward constructions of RTMs based on reversible elements than based on reversible logic gates. In particular, a remarkable feature of our constructions is that they can operate asynchronously, whereby the clock signal, which is indispensable for synchronizing all elements as in a reversible logic circuit, can possibly be removed from the construction.

© 2012 Elsevier B.V. All rights reserved.

## 1. Introduction

A reversible Turing machine (RTM) is a Turing machine of which all possible configurations constitute a linear graph without branching. In other words [19], a RTM's transition function is required to be bijective, such that the machine is unable to reach any certain configuration by transitions from different tape directions, or to reach a configuration from the same direction in two or more ways. Thus, from the current configuration (except the initial one), it can be uniquely determined what the previous configuration is, i.e., a RTM holds a kind of “backward determinism”.

It is known that reversible Turing machines can be built from various reversible technologies, e.g., reversible logic gate [8], the Billiard-Ball Model [17], reversible cellular automaton [22], DNA Polymer [26], etc. In addition, a reversible logic element resembles a reversible logic gate, in the sense that both of them comprise an equal number of input and output lines, except that the former also carries a 1-bit memory while the latter does not. The operation of a reversible element is determined by a bijective function between the inputs and outputs. Morita [21] proposed a reversible logic element, called the *Rotary Element* (RE), and demonstrated that the RE can be used to construct explicitly a certain RTM computing the function  $y = 2x$ . In particular, Morita's RTM consists of a Control module which serves as a finite-state machine of the RTM, connected to the right by an (infinite) array of Tape-Cell modules, each of which represents a unique cell on a (semi-infinite) tape. Here signals can be expressed as tokens, such that the binary values 0 and 1 of any input or output in conventional circuits are represented in terms of the absence and presence of a token on the corresponding line, respectively.

Although the RE enables a much simpler and more straightforward construction [21] of the RTM than to use reversible logic gates [4,9,18,27,28], there is an obvious disadvantage in Morita's construction with respect to computing time. Actually, for the sake of accomplishing an instantaneous transition of the RTM, a token must move back and forth within the construction composed by REs, between the Control module and a Tape-Cell module that corresponds to the cell being

\* Corresponding author. Tel.: +81 29 856 1420.

E-mail addresses: [lijia315yu@gmail.com](mailto:lijia315yu@gmail.com), [lijia@cqu.edu.cn](mailto:lijia@cqu.edu.cn) (J. Lee).

scanned by the RTM's head. The distance between the Control module and the Tape-Cell module at the destination, however, may become arbitrarily large and even unbounded (e.g., [11]), along with the progress of the computation. Thus, it makes sense to build an alternative construction of RTMs using reversible elements, which can work out any function in a much more efficient fashion.

In this paper, we propose a novel scheme for decomposing RTMs systematically into various reversible logic elements. For this purpose, instead of focusing on a specific RTM, our design scheme focuses on how to build a general reversible Turing machine out of reversible elements in a straightforward way. To realize a RTM, our scheme exploits a one-dimensional array of identical reversible modules that are interconnected locally and uniformly (see also [15]). In addition, each instantaneous transition of the RTM invokes at most one operation of a module at any time. To this extent, we show that the reversible module can be constructed using reversible logic elements with much simpler functionalities, involving a pair of mutually inverse elements which take a minimal number of input and output lines [12].

This paper is organized as follows: Sections 2 and 3 give overviews on reversible Turing machines and reversible logic elements, respectively. Section 4 describes our scheme for constructing a general RTM based on reversible elements. This paper finishes with the conclusions given in Section 5.

## 2. Reversible Turing machines

A *Turing machine* is a system which, in general, consists of an infinite one-dimensional tape divided into cells, and a read-write head that accesses a single cell on the tape at any time. Each cell is able to take a symbol from a finite set of symbols. At any time, the TM is in any one of a finite set of states, and executes a transition in accordance with the machine's current state and the symbol in the cell being scanned by the head.

Formally, a Turing machine can be defined as  $(Q, \Sigma, q_s, q_f, F)$  where  $Q$  and  $\Sigma$  are non-empty finite sets of states and tape symbols, respectively.  $q_s \notin Q$  is the start state and  $q_f \notin Q$  is the final (halt) state. For simplicity, let  $Q^{+s} = Q \cup \{q_s\}$  and  $Q^{+f} = Q \cup \{q_f\}$ . Moreover,  $F \subseteq Q \times \Sigma \times \Sigma \times Q \cup Q^{+s} \times \{/\} \times \{L, R\} \times Q^{+f}$  is a set of transitions. Assume  $a, b \in \Sigma$  and  $d \in \{L, R\}$ . A transition  $[p, a, b, q] \in F$  means that when the TM is in state  $p$  and reads symbol  $a$  from the cell currently being scanned by the head, the TM writes a new symbol  $b$  on the cell, and updates its state to  $q$ . For convenience, transitions given in such form are called *stationary*. Likewise, a transition  $[p, /, d, q] \in F$  expresses that if the TM is in state  $p$ , then the TM shifts its head one cell in the left (or right) direction when  $d = L$  (resp.  $d = R$ ), and changes its state to  $q$ . Hence, such transitions are called *moving transitions*.

Assume  $p \in Q^{+s}$  and  $q \in Q^{+f}$ . For convenience, let  $p \vdash q$  denote  $\exists(u, v) \in \Sigma^2 ([p, u, v, q] \in F)$  or  $\exists d \in \{L, R\} ([p, /, d, q] \in F)$ . Moreover, let  $p \vdash^* q$  represent that either  $p = q$ , or there are  $p_1, \dots, p_n \in Q$  ( $n \geq 0$ ) such that  $p \vdash p_1 \vdash \dots \vdash p_n \vdash q$ . Assume  $M_s = \{q \in Q \mid q_s \vdash^* q\}$  and  $M_f = \{q \in Q \mid q \vdash^* q_f\}$ . If  $M_s \subset Q$ , it is easy to verify that for each state  $q' \in Q \setminus M_s$ ,  $q'$  is unreachable from the state  $q_s$ . Similarly, for any state  $q'' \in Q \setminus M_f$ ,  $q''$  is unable to reach the final state  $q_f$ . Thus, it is reasonable to assume  $Q = M_s = M_f$ .

The quadruple form of transitions in  $F$  allows a straightforward definition on the reversibility of a TM [2,20]. Let  $\alpha_1$  and  $\alpha_2$  be two distinct transitions in  $F$ , i.e.,

$$\begin{aligned}\alpha_1 &= [p_1, u_1, v_1, q_1] \\ \alpha_2 &= [p_2, u_2, v_2, q_2].\end{aligned}$$

In this case,  $\alpha_1$  and  $\alpha_2$  is said to overlap in domain iff  $p_1 = p_2$  and the following (i) or (ii) holds:

- (i)  $u_1 = u_2$ ;
- (ii)  $u_1 = /$  or  $u_2 = /$ .

Moreover,  $\alpha_1$  and  $\alpha_2$  is said to overlap in range iff  $q_1 = q_2$  and the following (iii) or (iv) holds:

- (iii)  $v_1 = v_2$ ;
- (iv)  $v_1 = /$  or  $v_2 = /$ .

Thus, a TM  $(Q, \Sigma, q_s, q_f, F)$  is called *deterministic* if no pair of transitions in  $F$  overlap in domain. In addition, a deterministic TM is called *reversible* if no pair of transitions in  $F$  overlap in range.

From the above definition, a shift of the read-write head of a RTM depends solely on the local state but not on the symbol read from the tape. Morita et al. [20] showed that any arbitrary reversible Turing machine can be converted systematically into an equivalent RTM which employs merely two symbols, i.e.,  $\Sigma = \{0, 1\}$ . Besides, we further provide several conditions on the transitions of a RTM given in quadruple form, which can remarkably affect the efficient construction of the RTM using reversible logic elements. Let  $(Q, \Sigma, q_s, q_f, F)$  be such a RTM that holds the following conditions:

- (C1) For each  $p, q \in Q$  and  $a, b \in \Sigma$ ,

$$[p, a, b, q] \in F \implies a \neq b,$$

i.e., a stationary transition must change the tape symbol on a cell.

(C2) For each  $p_1, q_1 \in Q, q_2 \in Q^{+f}, a, b \in \Sigma$  and  $u, v \in \Sigma \cup \{/, L, R\}$ ,

$$[p_1, a, b, q_1] \in F \wedge [q_1, u, v, q_2] \in F \implies u = /.$$

That is, a stationary transition must be followed by a moving transition.

Since  $q_s \vdash^* q$  and  $q \vdash^* q_f$  for every  $q \in Q$ , condition C2 implies that for each stationary transition  $[p_1, a, b, p_2] \in F$ , there are moving transitions  $[q_1, /, d, p_1] \in F$  and  $[p_2, /, d', q_2] \in F$  where  $d, d' \in \{L, R\}$ . As described below, any RTM can be easily transformed into an equivalent RTM that satisfies all the above conditions.

Assume  $(Q, \Sigma, q_s, q_f, F)$  is a RTM. First of all, for each stationary transition  $\alpha_1 = [p_1, a, a, p_2]$  in  $F$ , we include a new state  $p'_1 (\notin Q)$  in  $Q$  and replace  $\alpha_1$  by transitions:  $[p_1, a, b, p'_1]$  and  $[p'_1, b, a, p_2]$  where  $b \in \Sigma$  and  $b \neq a$ . After that, for each  $p \in Q$ , let  $Q_r(p) = \{[p', a, b, p] \in F \mid a, b \in \Sigma\}$  and  $Q_l(p) = \{[p, a, b, p'] \in F \mid a, b \in \Sigma\}$ . Then, whenever  $Q_r(p) \neq \emptyset$  and  $Q_l(p) \neq \emptyset$ ,<sup>1</sup> we include two different states  $r(p) (\notin Q)$  and  $l(p) (\notin Q)$  in  $Q$  as well as two moving transitions:  $[r(p), /, R, l(p)]$  and  $[l(p), /, L, p]$  in  $F$ , followed by substituting each transition  $[p', a, b, p] \in Q_r(p)$  in  $F$  with transition  $[p', a, b, r(p)]$ . Hence, it is easy to verify that the resulting RTM works exactly as the original RTM, of which all transitions fulfill the conditions C1 and C2.

### 3. Reversible logic elements

A *logic element* is a module which consists of a finite number of input and output lines. Communication between an element and other elements is done via exchanging tokens through interconnection lines among them. The binary signal values 0 and 1 of conventional logic circuits can be encoded by the absence and presence of a token on a line, respectively.

A logic element can be formalized as  $(\mathcal{I}, \mathcal{O}, \Lambda, \lambda_0, \Psi)$ , where  $\mathcal{I}$  and  $\mathcal{O}$  are finite sets of input and output lines ( $\mathcal{I} \cap \mathcal{O} = \emptyset$ ), respectively.  $\Lambda$  is a finite set of internal states of the element, with  $\lambda_0 \in \Lambda$  being the initial state. In addition,  $\Psi \subseteq \mathcal{I} \times \Lambda \times \mathcal{O} \times \Lambda$  is a set of operations. Assume  $a \in \mathcal{I}, a' \in \mathcal{O}$ , and  $s, s' \in \Lambda$ . An operation  $a, s \rightarrow a', s' \in \Psi$  corresponds to a situation where the element, when in state  $s$  with a token appearing on its input line  $a$ , will assimilate the token from line  $a$ , then generates a token on output line  $a'$  and changes its state to  $s'$ .

Assume a logic element  $(\mathcal{I}, \mathcal{O}, \Lambda, \lambda_0, \Psi)$ . Let

$$\begin{aligned} \theta_1 &= a_1, s_1 \rightarrow a'_1, s'_1 \\ \theta_2 &= a_2, s_2 \rightarrow a'_2, s'_2 \end{aligned}$$

be any two operations in  $\Psi$  ( $\theta_1$  and  $\theta_2$  may be the same). The element is called *deterministic* iff

$$a_1 = a_2 \wedge s_1 = s_2 \implies \theta_1 = \theta_2.$$

Moreover, the element is *reversible* iff it is deterministic, and

$$a'_1 = a'_2 \wedge s'_1 = s'_2 \implies \theta_1 = \theta_2.$$

Thus, each operation of a reversible element does not overlap on the left-hand side or on the right-hand side with any other operations.

Apparently, reversing the input and output lines, as well as the left-hand and right-hand sides of each operation of a reversible logic element gives rise to another reversible element, which has the inverse functionality of the former (e.g. [24]). In this paper, we focus on reversible elements each of which has a minimal number of states, i.e.,  $|\Lambda| = 2$ . Fig. 1 gives some useful reversible elements, among which the RT and IRT are mutually inverse to each other, so as the RD and IRD. In addition, it can be verified that the inverse element of an RE is still an RE. The case for an CDR is the same.

### 4. Realization of reversible Turing machines by reversible logic elements

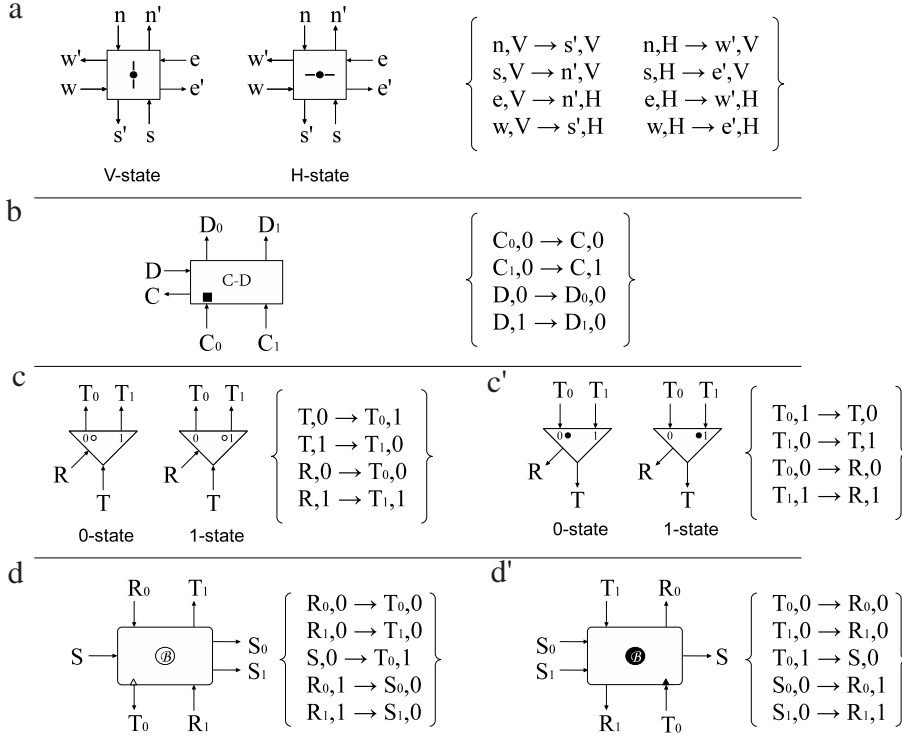
#### 4.1. A novel reversible element capable of realizing reversible Turing machines

Let  $\mathcal{A} = (Q, \{0, 1\}, q_s, q_f, F)$  be a reversible Turing machine that satisfies both the conditions C1 and C2 given in Section 2. For convenience, let  $M_R, M_L, S_D, S_R$  be subsets of  $Q$  defined as follows:

$$\begin{aligned} M_R &= \{q \in Q \mid [p, /, R, q] \in F\} \\ M_L &= \{q \in Q \mid [p, /, L, q] \in F\} \\ S_R &= \{q \in Q \mid [p, s, t, q] \in F \wedge s, t \in \{0, 1\}\} \\ S_D &= \{p \in Q \mid [p, s, t, q] \in F \wedge s, t \in \{0, 1\}\}. \end{aligned}$$

As mentioned in Section 2, it is obvious that  $S_D \subseteq M_L \cup M_R$  due to the condition C2. Let  $M_R^\# = M_R \setminus S_D$  and  $M_L^\# = M_L \setminus S_D$ , i.e.,  $M_R^\#$  and  $M_L^\#$  contain those states in either of which the RTM's head, when moving to a cell in the right or left direction,

<sup>1</sup> If  $Q_r(p) \neq \emptyset$  and  $Q_l(p) \neq \emptyset$ , then  $[p, /, d, q] \notin F$  and  $[q, /, d, p] \notin F$  for any  $q \in Q \cup \{q_s, q_f\}$  and  $d \in \{L, R\}$ , due to the reversibility of transitions.



**Fig. 1.** Reversible logic elements and their respective sets of operations. (a) **RE** (Rotary Element) =  $(\{n, s, e, w\}, \{n', s', e', w'\}, \{V, H\}, V, \Psi_{RE})$ : The local states  $H$  and  $V$  are displayed by horizontal and vertical bars, respectively. Roughly speaking, when a token comes from a direction parallel to the bar, then the token will pass straight through to the opposite output line without changing the direction of the bar, e.g., from  $n$  to  $s'$  when the RE is in state  $V$ . When a token comes from a direction orthogonal to the bar, the token will be deflected to the right and rotate the bar  $90^\circ$ , e.g., from  $n$  to  $w'$  when the RE is in state  $H$ . (b) **CDE** (Coding-Decoding Element) =  $(\{C_0, C_1, D\}, \{D_0, D_1, C\}, \{0, 1\}, 0, \Psi_{CD})$ : Assume  $i \in \{0, 1\}$ . When the element is in state 0, a token arriving on input line  $C_i$  updates the state to  $i$ , and gives rise to a token on output line  $C$ . A token arriving on input line  $D$  reverts the state to 0, and results in a token on output line  $D_i$  if the element is in state  $i$ . (c) **RT** (Reading Toggle) =  $(\{T, R\}, \{T_0, T_1\}, \{0, 1\}, 0, \Psi_{RT})$ : When the element is in state 0 (or 1), a token arriving on input line  $T$  changes the state to 1 (resp. 0), and results in a token on output line  $T_0$  (resp.  $T_1$ ); a token arriving on input line  $R$  does not change the state and gives rise to a token on output line  $T_0$  (resp.  $T_1$ ). (c') **IRT** (Inverse Reading Toggle) =  $(\{T_0, T_1\}, \{T, R\}, \{0, 1\}, 0, \Psi_{IRT})$ : When the element is in state 0 (or 1), a token arriving on input line  $T_1$  (resp.  $T_0$ ) changes the state to 1 (resp. 0), and results in a token on output line  $T$ ; a token arriving on input line  $T_0$  (resp.  $T_1$ ) does not change the state, and gives rise to a token on output line  $R$ . (d) **RD** (Redirector) =  $(\{S, R_0, R_1\}, \{T_0, T_1, S_0, S_1\}, \{0, 1\}, 0, \Psi_{RD})$ : Assume  $i \in \{0, 1\}$ . When the element is in state 0, a token arriving on input line  $R_i$  does not change the state, and results in an output token on line  $T_i$ ; an input token on line  $S$  changes the state to 1, and gives rise to a token on output line  $T_0$ . When the element is in state 1, a token arriving on input line  $R_i$  reverts the state to 0, and results in an output token on line  $S_i$ . (d') **IRD** (Inverse Redirector) =  $(\{T_0, T_1, S_0, S_1\}, \{S, R_0, R_1\}, \{0, 1\}, 0, \Psi_{IRD})$ : Assume  $i \in \{0, 1\}$ . When the element is in state 0, a token arriving on input line  $T_i$  does not change the state, and results in an output token on line  $R_i$ ; an input token on line  $S_i$  changes the state to 1, and gives rise to a token on output line  $R$ . When the element is in state 1, a token arriving on input line  $T_0$  reverts the state to 0, and results in an output token on line  $S$ .

will simply go on shifting without accessing the symbol on that cell. Moreover, because  $S_R \cap (M_R \cup M_L) = \emptyset$  and  $q_s \vdash^* q$  for each  $q \in Q$ , we obtain  $Q = M_L \cup M_R \cup S_R$ .

For simplicity, let  $N(L)$ ,  $N(R)$ ,  $n(L)$ ,  $n(R)$  be integers and  $d \in \{L, R\}$ , such that  $N(d) = |M_d|$  and  $n(d) = |M_d \cap S_D|$ . Without loss of generality, assume that  $M_d \cap S_D = \{q_1^d, q_2^d, \dots, q_{n(d)}^d\}$  and  $M_d^\# = \{q_{n(d)+1}^d, \dots, q_{N(d)}^d\}$ .

The RTM  $\mathcal{A}$  can be simulated by a one-dimensional array of identical reversible elements, called RCU (Reversible Cell Unit), that are interconnected locally in a uniform way. Here an RCU is formulated as  $(\mathcal{I}_A, \mathcal{O}_A, \{0, 1\}, 0, \Psi_A)$ , (see Fig. 2), where

$$\begin{aligned} \mathcal{I}_A &= \{I_1^L, \dots, I_{N(L)}^L, I_1^R, \dots, I_{N(R)}^R\} \cup \{\text{begin}\}, \\ \mathcal{O}_A &= \{O_1^L, \dots, O_{N(L)}^L, O_1^R, \dots, O_{N(R)}^R\} \cup \{\text{end}\}. \end{aligned}$$

Also, the set of operations  $\Psi_A$  is defined in the following way.

- (i) For each  $c, d \in \{L, R\}$ ,  $1 \leq j \leq n(d)$ ,  $1 \leq k \leq N(c)$ ,  $s, t \in \{0, 1\}$  ( $s \neq t$ ) and  $p' \in S_R$ , if  $[q_j^d, s, t, p'] \in F$  and  $[p', /, c, q_k^c] \in F$  (condition C2), then include the next operation in  $\Psi_A$ :

$$I_j^d, s \rightarrow O_k^c, t.$$

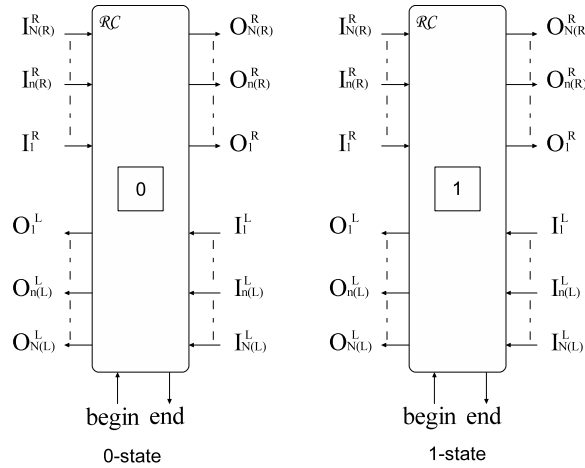


Fig. 2. Two RCUs.

(ii) For every  $d, e \in \{L, R\}$ ,  $n(d) < i \leq N(d)$ , and  $1 \leq l \leq N(e)$ , if  $[q_i^d, /, e, q_l^e] \in F$  (i.e.,  $q_i^d \in M_d^\#$ ), then for each  $s \in \{0, 1\}$ , include the following operation to  $\Psi_A$ :

$$I_i^d, s \rightarrow O_l^e, s.$$

(iii) Suppose  $d \in \{L, R\}$ ,  $1 \leq j \leq n(d)$ ,  $n(d) < i \leq N(d)$ ,  $s, t \in \{0, 1\}$  ( $s \neq t$ ) and  $p' \in S_R$ . If  $[q_j^d, s, t, p'] \in F$  and  $[p', /, c, q_f] \in F$ , then include the next operation in  $\Psi_A$ :

$$I_j^d, s \rightarrow \text{end}, t;$$

otherwise, if  $[q_i^d, /, e, q_f] \in F$ , for each  $w \in \{0, 1\}$ , include the following operation to  $\Psi_A$ :

$$I_i^d, w \rightarrow \text{end}, w.$$

(iv) Assume  $d \in \{L, R\}$  and  $1 \leq i \leq N(d)$ . If  $[q_s, /, d, q_i^d] \in F$ , then for each  $t \in \{0, 1\}$ , include the following operation to  $\Psi_A$ :

$$\text{begin}, t \rightarrow O_i^d, t.$$

In order to verify the RCU's reversibility, assume

$$\tau_1 = I_i^d, s_1 \rightarrow O_k^e, t$$

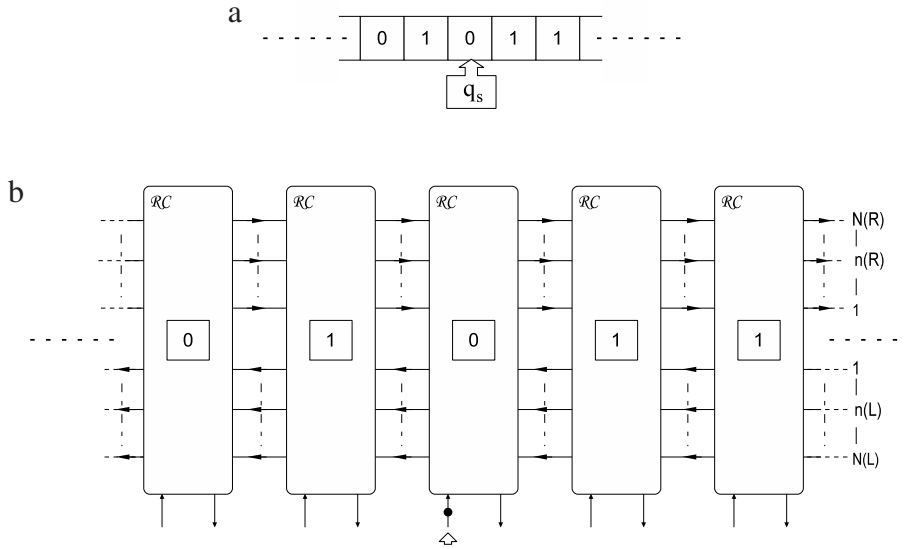
$$\tau_2 = I_j^c, s_2 \rightarrow O_k^e, t$$

are two different operations in  $\Psi_A$  whose right-hand sides coincident each other (the case for overlapping on left-hand side is similar). First of all, suppose  $n(d) < i \leq N(d)$  and  $n(c) < j \leq N(c)$ , whereby  $\tau_1$  and  $\tau_2$  are uniquely produced from the transitions  $[q_i^d, /, e, q_k^e]$  and  $[q_j^c, /, e, q_k^e]$  in  $F$ , respectively. The reversibility of the RTM  $\mathcal{A}$ , therefore, results in  $q_i^d = q_j^c$ , and hence,  $I_i^d = I_j^c$ . Combined with  $s_1 = s_2 = t$  (see (ii) above), this contradicts with the assumption that  $\tau_1 \neq \tau_2$ . Next, assume  $1 \leq i \leq n(d)$  for convenience, i.e., there is  $p \in S_R$  such that  $[q_i^d, s_1, t, p] \in F$  ( $s_1 \neq t$ ) and  $[p, /, e, q_k^e] \in F$ . In this case, because  $S_R \cap M_c = \emptyset$ , it is obvious that  $[q_j^c, /, e, q_k^e] \notin F$  for any  $1 \leq j \leq N(c)$ . Hence, we can only have  $1 \leq j \leq n(c)$  and  $[q_j^c, s_2, t, p] \in F$ . The reversibility between transitions  $[q_i^d, s_1, t, p]$  and  $[q_j^c, s_2, t, p]$ , however, yields  $q_i^d = q_j^c$  as well as  $s_1 = s_2$ , and thus  $\tau_1 = \tau_2$ , which still contradicts with the assumption of the above.

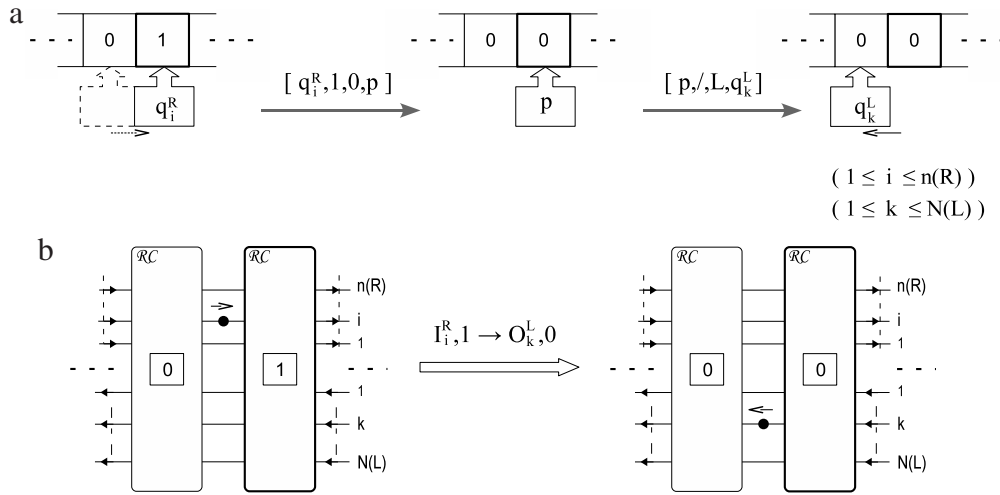
Furthermore, all other cases can be verified in the similar manner. In conclusion, because of the reversibility of all transitions in  $F$  (see Section 2), no pair of operations in  $\Psi_A$  overlap on left-hand side or on right-hand side, thereby our RCU is reversible.

Fig. 3 gives an initial configuration of the RTM  $\mathcal{A}$ , beneath which a one-dimensional array of RCUs that are uniformly interconnected is illustrated. In particular, the correspondence between the RCU array and cells on the tape of  $\mathcal{A}$  is one-to-one, whereby the symbol on each cell is encoded by the internal state of a corresponding TCU.

To see how the circuit in Fig. 3(b) works, for example, assume  $c, d \in \{L, R\}$ ,  $1 \leq i \leq n(d)$ ,  $1 \leq j \leq N(c)$ ,  $s, t \in \{0, 1\}$  and  $p \in S_R$ , such that  $[q_i^d, s, t, p] \in F$  and  $[p, /, c, q_j^c] \in F$  (i.e.,  $I_i^d, s \rightarrow O_j^c, t \in \Psi_A$ ). Also, suppose that the RTM  $\mathcal{A}$  is in state  $q_i^d$  whose head scans a cell with symbol  $s$ . In this case, the RCU corresponding uniquely to the cell being scanned by  $\mathcal{A}$ 's head is in state  $s$ , and it receives a single token arriving on its input line  $I_i^d$ , which is used to encode the state  $q_i^d$  of  $\mathcal{A}$ , as demonstrated in Fig. 4. Hence, according to the operation  $I_i^d, s \rightarrow O_j^c, t$ , the RCU is activated to assimilate the input token, change its state to  $t$ , and finally output a token on line  $O_j^c$ . The resulting token, therefore, will be transferred to input line  $I_j^c$  of the next RCU in direction  $c$ , which depicts that  $\mathcal{A}$  shifts its head to a neighboring cell in direction  $c$  and changes its state



**Fig. 3.** (a) An initial configuration of reversible Turing machine  $\mathcal{A}$ . (b) One-dimensional array of RCUs, in which each output line  $O_j^d$  of an RCU where  $d \in \{L, R\}$  and  $1 \leq j \leq N(d)$  is connected to the input line  $I_j^d$  of a neighboring RCU in direction  $d$ . Moreover, each RCU corresponds uniquely to a square cell on the tape of  $\mathcal{A}$ , whose state has been set beforehand to equal to the symbol on that cell. Here a token, represented by a black blob, is assigned on the input line *begin* of an RCU which corresponds to the cell being scanned by the RTM  $\mathcal{A}$ 's head at the initial time.



**Fig. 4.** (a) A stationary transition of RTM  $\mathcal{A}$  on a cell, which is followed immediately by a moving transition to leave that cell. (b) Simulation of the above instantaneous transitions by an operation of the corresponding RCU.

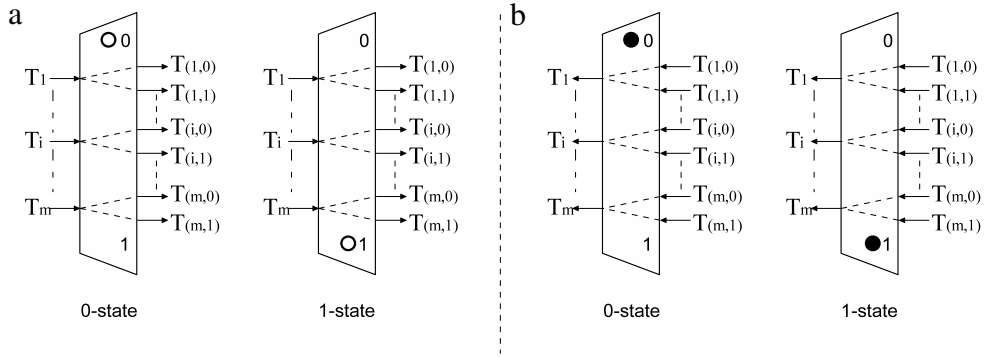
to  $q_f^c$ . The case of pure moving transition on a cell can be verified easily in a similar manner. To this extent, each instantaneous transition of  $\mathcal{A}$  on a cell can be exactly emulated by at most one operation of an RCU corresponding to that cell.

As shown in Fig. 3(b), simulation of  $\mathcal{A}$  is started by assigning a token on input line *begin* of a TCU which corresponds to the cell initially scanned by the RTM's head. After that, the token will run around in the array, each time moving from one RCU to a neighboring RCU in the left or right direction, which actually simulates  $\mathcal{A}$  working on individual cells, as described above. The whole process will be finished once an RCU in the array outputs a token to its line *end*, i.e.  $\mathcal{A}$  reaches the final state  $q_f$ , whereby the RTM halts.

#### 4.2. Decomposing RCU into simpler reversible elements

Now we proceed to decompose an RCU into much simpler reversible elements, like those elements in Fig. 1. For this purpose, let

$$\varphi : S_R \cup M_L^\# \cup M_R^\# \cup \{q_s\} \rightarrow M_L \cup M_R \cup \{q_f\}$$



**Fig. 5.** (a) **m-Toggle** =  $(\{T_i \mid 1 \leq i \leq m\}, \{T(k, s) \mid 1 \leq k \leq m \wedge 0 \leq s \leq 1\}, \{0, 1\}, 0, \Psi_{mt})$  where  $m > 0$  and  $\Psi_{mt} = \{T_i, s \rightarrow T(i, s), t \mid 1 \leq i \leq m \wedge 0 \leq s, t \leq 1 \wedge s \neq t\}$ . Each input line  $T_i$  of an m-Toggle is associated individually with two output lines  $T(i, 0)$  and  $T(i, 1)$ , on which the arrival of a token changes the state to 1 (or 0) when the element is in state 0 (resp. 1), and results in an output token on line  $T(i, 0)$  (resp.  $T(i, 1)$ ). (b) **Inverse m-Toggle** =  $(\{T(k, s) \mid 1 \leq k \leq m \wedge 0 \leq s \leq 1\}, \{T_i \mid 1 \leq i \leq m\}, \{0, 1\}, 0, \Psi_{imt})$  where  $m > 0$  and  $\Psi_{imt} = \{T(i, s), t \rightarrow T_i, s \mid 1 \leq i \leq m \wedge 0 \leq s, t \leq 1 \wedge s \neq t\}$ . Each output line  $T_i$  of an m-Toggle is associated individually with two input lines  $T(i, 0)$  and  $T(i, 1)$ . When the element is in state 0 (or 1), the arrival of a token on input line  $T(i, 1)$  (resp.  $T(i, 0)$ ) changes the state to 1 (resp. 0), and finally results in a token on output line  $T_i$ .

be a mapping defined as follows: For each  $p \in S_R \cup M_L^\# \cup M_R^\# \cup \{q_s\}$ ,  $q \in M_L \cup M_R \cup \{q_f\}$  and  $d \in \{L, R\}$ , we obtain

$$[p, /, d, q] \in F \iff \varphi(p) = q.$$

For every  $q \in M_L \cup M_R \cup \{q_f\}$ , there are, by definition,  $p \in Q^{+s}$  and  $d \in \{L, R\}$  such that  $[p, /, d, q] \in F$ . Thus, we obtain  $p \notin S_D$ , and hence,  $p \in S_R \cup M_L^\# \cup M_R^\# \cup \{q_s\}$  since  $S_D \subseteq M_L \cup M_R$  and  $Q^{+s} = S_R \cup M_L \cup M_R \cup \{q_s\}$ . Thus, the mapping  $\varphi$  is surjective. Furthermore, because  $p \vdash^* q_f$  for each  $p \in S_R \cup M_L^\# \cup M_R^\# \cup \{q_s\}$  together with the condition C2 in Section 2, the mapping  $\varphi$  is an injective function due to reversibility of the transitions in  $F$ . Combined with its surjectivity, the function  $\varphi$ , therefore, is bijective.

Additionally, let  $\rho$  be a mapping between  $M_L \cup M_R \cup \{q_f\}$  and  $\mathcal{O}_A$  the set of output lines of an RCU, such that  $\rho(q_f) = O_f$  and  $\rho(q_i^d) = O_i^d$  for every  $d \in \{L, R\}$  and  $1 \leq i \leq N(d)$ . Obviously, the mapping  $\rho$  is bijective. Thus, we can obtain a composite function  $\rho \circ \varphi : S_R \cup M_L^\# \cup M_R^\# \cup \{q_s\} \rightarrow \mathcal{O}_A$  and let  $\Gamma = \rho \circ \varphi$ , which of course is bijective.

For convenience, denote  $S_R$  by  $\{p_1^s, p_2^s, \dots, p_{N(S)}^s\}$  where  $N(S) = |Q| - N(L) - N(R) = n(L) + n(R)$ . Let  $\mu : S_D \times \{0, 1\} \rightarrow \{0, 1\} \times S_R$  be a function such that for each  $d \in \{L, R\}$ ,  $1 \leq i \leq n(d)$ ,  $1 \leq j \leq N(S)$  and  $s, t \in \{0, 1\}$ , we define

$$[q_i^d, s, t, p_j^s] \in F \iff \mu(q_i^d, s) = (t, p_j^s).$$

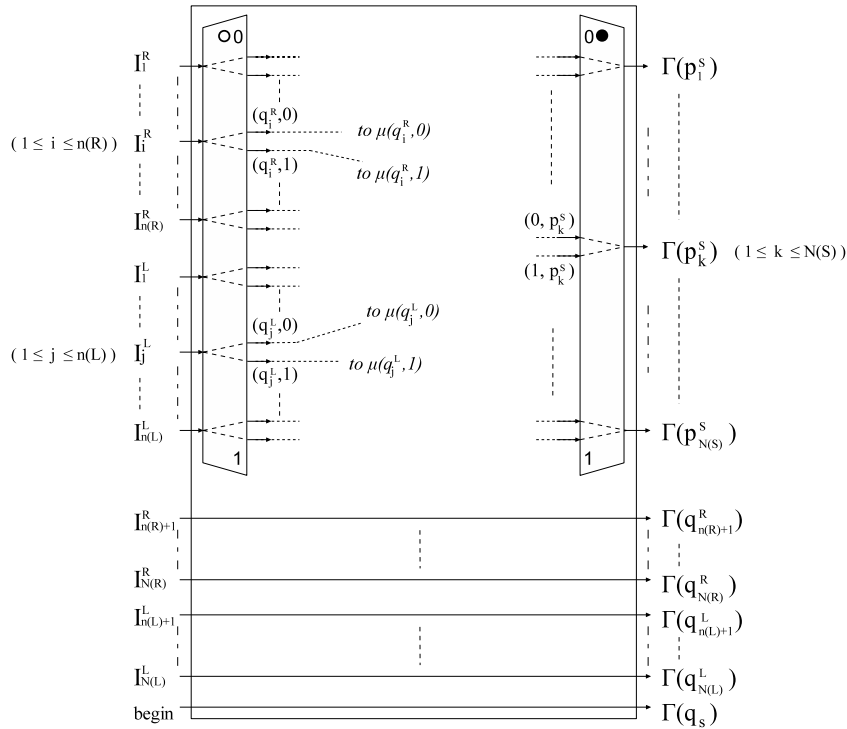
Fig. 5 gives two reversible elements, called m-Toggle and Inverse m-Toggle, that have mutually inverse functionalities. Based upon these elements, we can realize an RCU in a straightforward way, as illustrated in Fig. 6. Assume  $d \in \{L, R\}$ . Each input line  $I_i^d \in \mathcal{I}_A$  of the RCU in Fig. 6 with  $n(d) < i \leq N(d)$  is directly connected to an output line  $\Gamma(q_i^d)$ , so as the input line *begin* being connected to  $\Gamma(q_s)$ . Moreover, each input line  $I_i^d \in \mathcal{I}_A$  where  $1 \leq i \leq n(d)$  is directed to an input line of the m-Toggle, which is associated with two output lines  $(q_i^d, 0)$  and  $(q_i^d, 1)$  of the m-Toggle. Likewise, each output line  $\Gamma(p_k^s) \in \mathcal{O}_A$  where  $1 \leq k \leq N(S)$  also serves as an output line of the Inverse m-Toggle, that is associated with two input lines  $(0, p_k^s)$  and  $(1, p_k^s)$  of the Inverse m-Toggle. To this extent, for each  $d \in \{L, R\}$ ,  $1 \leq i \leq n(d)$ ,  $1 \leq k \leq N(S)$ ,  $s, t \in \{0, 1\}$  and  $s \neq t$ , an output line  $(q_i^d, s)$  of the m-Toggle is connected to an input line  $(t, p_k^s)$  of the Inverse m-Toggle, iff  $\mu(q_i^d, s) = (t, p_k^s)$  (i.e.,  $[q_i^d, s, t, p_k^s] \in F$ ).

To verify the validness of the construction in Fig. 6, for example, suppose  $c, d \in \{L, R\}$ ,  $1 \leq i \leq n(d)$ ,  $1 \leq k \leq N(S)$ ,  $1 \leq j \leq N(c)$ , and  $s, t \in \{0, 1\}$  ( $s \neq t$ ), such that  $[q_i^d, s, t, p_k^s] \in F$  and  $[p_k^s, /, c, q_j^c] \in F$ , and hence,  $I_i^d, s \rightarrow O_j^c, t \in \Psi_A$ . In addition, assume both the m-Toggle and Inverse m-Toggle are in state  $s$ , and a token arrives on input line  $I_i^d$  of the construction. In this case, the token changes the state of the m-Toggle from  $s$  to  $t$ , and gives rise to a token on line  $(q_i^d, s)$  that will be fed to the input line  $(t, p_k^s)$  of the Inverse m-Toggle because  $\mu(q_i^d, s) = (t, p_k^s)$ . As a result, the token also changes the Inverse m-Token from state  $s$  to state  $t$ , and finally results in a token on output line  $\Gamma(p_k^s) = O_j^c$ . Thus, it can be verified that the construction in Fig. 6 actually behaves as an RCU.

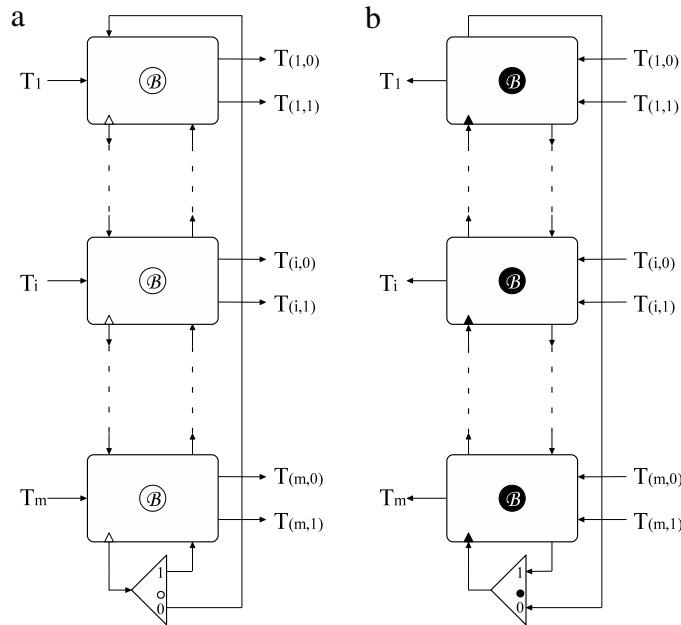
Moreover, Fig. 6 stipulates that for the sake of decomposing an RCU, it is enough to realize an m-Toggle as well as an Inverse m-Toggle by means of much simpler reversible elements. To this end, Fig. 7 shows the construction of an m-Toggle using RD and 1-Toggle (or simply Toggle), along with an Inverse m-Toggle realized by IRD and Inverse Toggle. The correctness of these constructions can be easily verified, for example, as demonstrated in Fig. 8. In addition, since both RD and Toggle, as well as their respective inverses, can be easily realized from the RE (see Fig. 9), it is able to obtain the RE-based construction of an m-Toggle through taking place of each RD as well as the Toggle in Fig. 7(a) by an RE. The case for an Inverse m-Toggle is similar.

As stated before, the simplest reversible elements include the RT and IRT, which have merely two input and two output lines. Obviously, the RT and IRT actually cover the functionalities of a Toggle and Inverse Toggle, respectively. In addition,



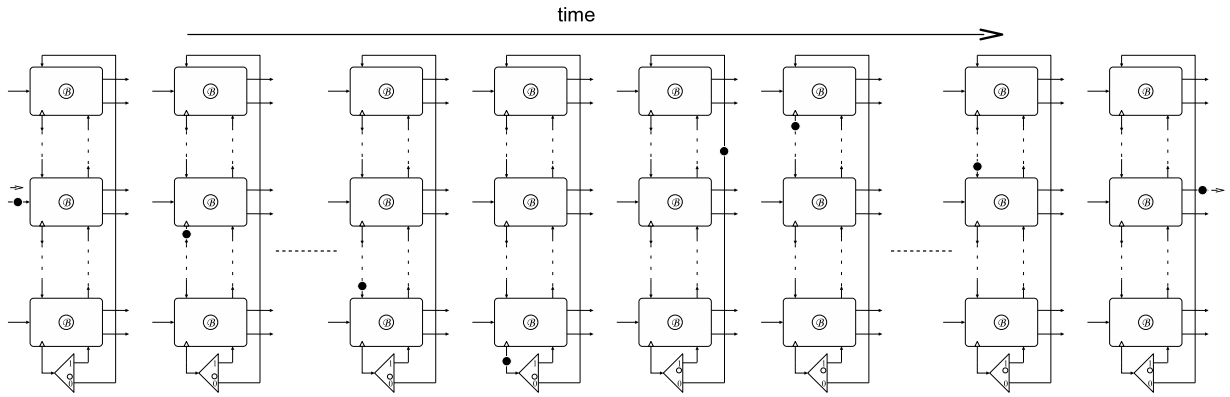


**Fig. 6.** Construction of an RCU  $(J_A, \mathcal{O}_A, \{0, 1\}, \Psi_A)$  by m-Toggle and Inverse m-Toggle where  $m = n(R) + n(L) \equiv N(S)$ . For convenience, positions of an RCU's input and output lines in Fig. 2 are relocated here, and  $\mathcal{O}_A$  is denoted in terms of  $\{\Gamma(q) \mid q \in S_R \cup M_L^\# \cup M_R^\# \cup \{q_s\}\}$  with  $\Gamma$  being a bijective mapping from  $S_R \cup M_L^\# \cup M_R^\# \cup \{q_s\}$  to  $\mathcal{O}_A$ . In addition, an RCU's local states 0 and 1 are encoded, respectively, by both the m-Toggle and Inverse m-toggle being in state 0 and state 1 at the same time.

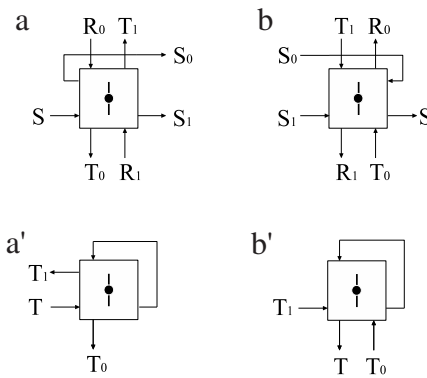


**Fig. 7.** (a) Realization of m-Toggle ( $m > 0$ ) using RD and Toggle. Here an m-Toggle's local states 0 and 1 are represented by the internal states 0 and 1 of the Toggle, respectively. (b) Construction of Inverse m-Toggle by IRD and Inverse Toggle. Due to the reversibility of all logic elements, the Inverse m-Toggle is obtained by simply reversing the directions of all input and output lines, as well as each element's logic functionality in the left construction of an m-Toggle.

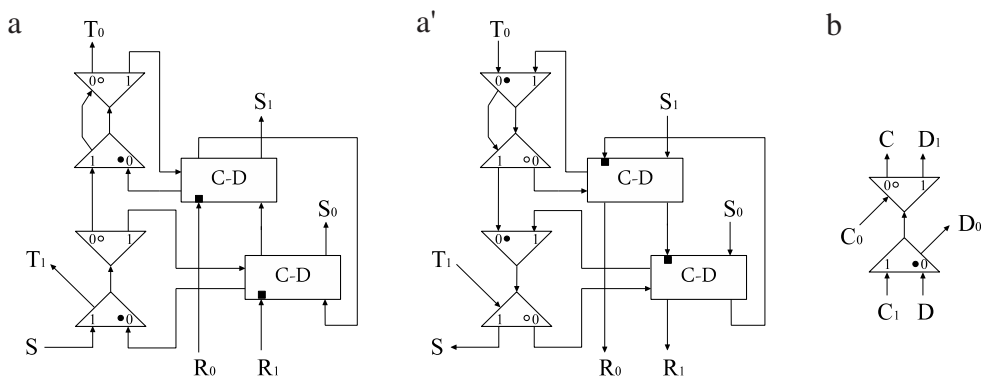




**Fig. 8.** Trace of a token running around in the construction of an m-Toggle in Fig. 7(a). Roughly speaking, suppose a token arrives on an input line  $T_i$  where  $1 \leq i \leq m$ . This token will be fed to input line  $S$  of the  $i$ -th RD from the top, thereby it will change the RD's state to 1 and be transferred to output line  $T_0$  of the RD, whereas all other RDs remain in state 0 (the initial state). After that, the token will eventually arrive at the Toggle at the bottom, change its state from 0 (or 1) to 1 (resp. 0) if the (m-)Toggle is in state 0 (resp. 1), and finally return to the  $i$ -th RD from the element's input line  $R_0$  (resp.  $R_1$ ). The token, therefore, will revert the  $i$ -th RD's state to 0, and give rise to an output on line  $T(i, 0)$  (resp.  $T(i, 1)$ ). Thus, it is easy to verify that the construction works exactly in the same way as an m-Toggle.



**Fig. 9.** Realizations of (a) RD and (a') IRD using RE. (b) Realization of (1-)Toggle by RE. For simplicity, the output lines  $T(1, 0)$  and  $T(1, 1)$  of a Toggle are abbreviated to  $T_0$  and  $T_1$ , respectively. (b') Similarly, construction of Inverse Toggle by RE.



**Fig. 10.** Realizations of (a) RD and (a') IRD using RT, IRT and CDE, in either of which all RTs and IRTs being in state 0 (or 1) represents that the RD or IRD is in state 0 (resp. 1). Like the Inverse m-Toggle in Fig. 7, an IRT is obtained by exchanging the input and output lines, as well as reversing the functionality of each element in an RT's construction, except that the inverse of a CDE remains a CDE. (b) Construction [12] of CDE by RT and IRT.

though Fig. 9 suffices to prove that an RD and IRD can be constructed from RT and IRT [12], both constructions can be made much more efficient by using these elements directly, as shown in Fig. 10, rather than based on the construction of an RE by RT and IRT. Moreover, a typical sequence of the RD processing an input token is illustrated in Fig. 11, via which the correctness of the construction in Fig. 10 can be easily verified. As a result, the RT and IRT can be used to construct an m-Toggle as well as an Inverse m-Toggle, and so as the RCU.

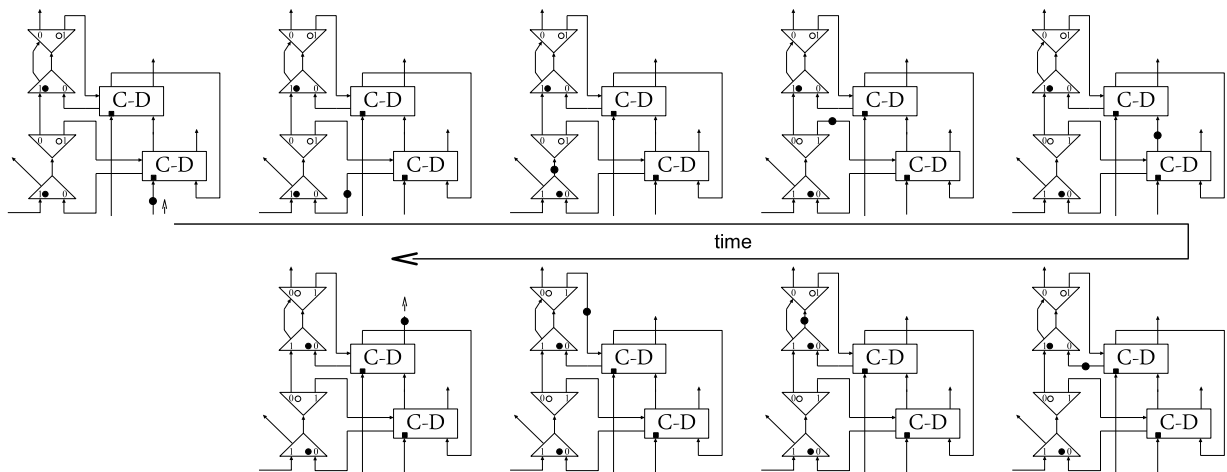


Fig. 11. Trace of a token arriving on input line  $R_1$  of an RD's construction in Fig. 10(a), when the RD is in state 1.

## 5. Conclusions

In this paper, we presented a new scheme for constructing reversible Turing machines with one tape and two symbols based on reversible logic elements. For this purpose, we imposed several conditions on the transitions of RTMs, which allow for decomposing any RTM into a network of reversible elements efficiently in a systematical way. In particular, our construction is composed by a linear chain of identical reversible modules that are locally and uniformly interconnected with each other, resembling a one-dimensional cellular array. Such regular and uniform construction of a RTM not only can account for the profound overhead in computing time caused by the heterogeneous structure of the previous construction [21], but also might contribute to the potential of array-structured computers [1,25]. Furthermore, each module in the construction can be built from various reversible elements each of which has much simpler functionality than the module.

Because at most one token will run around in the construction of a RTM at all times, delays involved in the operation of any element or in the transmission of the token on any line do not disturb the correct computation of the RTM [21,12]. Thus, our construction of a RTM can operate in a clockless (asynchronous) mode (see also [14]). Reversibility not only promises in principle the possibility of computers with zero dissipation [2,5,7,16,23], but is also essential for quantum [6] and fluctuation-based [3,13,14] computing. As with reversibility, asynchronicity tends to reduce power consumption and energy dissipation [10], it is yet due to different reasons: Though reversible systems reduce power consumption by preventing entropy loss, reversible systems do so by keeping all idle elements in a sleeping state.

Finally, although 1-tape, 2-symbol reversible Turing machines are indeed universal [20], it might make sense to conceive the construction of more general RTMs that can take an arbitrary number of tape symbols. Nevertheless, such construction may possibly be achieved by replacing each RCU module in the construction in Fig. 3(b) with a reversible component, provided that the component contains more internal states and more complicated operations. The use of more states, on the other hand, tends to hinder the effective decomposition of the component into reversible logic elements with 1-bit memory. As a detailed analysis of this respect is beyond the scope of this paper, further investigations are left for future studies.

## Acknowledgments

We are grateful to the anonymous reviewer for the helpful comments and valuable suggestions. We also thank Dr. Ferdinand Peper at National Institute of Information and Communications Technology, Japan, for the fruitful discussions. This research work was supported by NSFC (61170036).

## References

- [1] A. Bandyopadhyay, R. Pati, S. Sahu, F. Peper, D. Fujita, Massively parallel computing on an organic molecular layer, *Nat. Phys.* 6 (2010) 369–375.
- [2] C.H. Bennett, Logical reversibility of computation, *IBM J. Res. Dev.* 17 (1973) 525–532.
- [3] C.H. Bennett, The thermodynamics of computation—a review, *Int. J. Theor. Phys.* 21 (12) (1982) 905–940.
- [4] A. Brodsky, Reversible circuit realizations of Boolean functions, in: *Proc. 3rd IFIP Int. Conf. on Theoretical Computer Science*, Toulouse, France, 2004.
- [5] H. Buhrman, J. Tromp, P. Vitányi, Time and space bounds for reversible simulation, *J. Phys. A: Math. Gen.* 34 (2001) 6821–6830.
- [6] F. Feynman, Quantum mechanical computers, *Found. Phys.* 16 (6) (1986) 507–531.
- [7] M. Frank, T. Knight, N. Margolus, Reversibility in optimally scalable computer architectures, in: *Proc. 1st Int. Conf. on Unconventional Models of Computation*, Springer, Berlin, 1998.
- [8] E. Fredkin, T. Toffoli, Conservative logic, *Int. J. Theor. Phys.* 21 (3–4) (1982) 219–253.
- [9] P. Gupta, A. Agrawal, N.K. Jha, An algorithm for synthesis of reversible logic circuits, *IEEE Trans. CAD* 25 (11) (2006) 2317–2330.
- [10] S. Hauck, Asynchronous design methodologies: an overview, *Proc. IEEE* 83 (1) (1995) 69–93.

- [11] J.E. Hopcroft, J.D. Ullman, Introduction to Automata Theory, Languages, and Computation, Addison-Wesley, Massachusetts, 1979.
- [12] J. Lee, F. Peper, S. Adachi, K. Morita, An asynchronous cellular automaton implementing 2-state 2-input 2-output reversed-twin reversible elements, in: ACRI 2008, in: LNCS, vol. 5191, 2008, pp. 67–76.
- [13] J. Lee, A simple model of asynchronous cellular automata exploiting fluctuation, *J. Cell. Autom.* 6 (2011) 341–352.
- [14] J. Lee, K. Imai, Q.S. Zhu, Fluctuation-driven computing on number-conserving cellular automata, *Inf. Sci.* 187 (2012) 266–276.
- [15] J. Lee, Q.S. Zhu, A direct proof of Turing universality of delay-insensitive circuits, *Int. J. Unconv. Comput.* 8 (2) (2012) 107–118.
- [16] M. Li, J. Tromp, P. Vitányi, Reversible simulation of irreversible computation, *Physica D* 120 (1998) 168–176.
- [17] N. Margolus, Physics-like models of computation, *Physica D* 10 (1984) 81–95.
- [18] D. Maslov, G.W. Dueck, D.M. Miller, Techniques for the synthesis of reversible Toffoli networks, *ACM Trans. Des. Autom. Electron. Syst.* 12 (4) (2007).
- [19] A.B. Matos, A Turing machine model where the duality between determinism and reversibility is explicit, DCC-FCUP Internal Report, Universidade do Porto, Portugal, 2001.
- [20] K. Morita, A Shirasaki, Y. Gono, A 1-tape 2-symbol reversible Turing machine, *Trans. IEICE* 72 (3) (1989) 223–228.
- [21] K. Morita, A simple universal logic element and cellular automata for reversible computing, in: Proc. 3rd Int. Conference on Machines, Computations, and Universality, Chisinau, in: LNCS, vol. 2055, 2001, pp. 102–113.
- [22] K. Morita, Y. Tojima, K. Imai, A simple computer embedded in a reversible and number-conserving two-dimensional cellular space, *Multi. Val. Logic* 6 (2001) 483–514.
- [23] K. Morita, Reversible computing and cellular automata — a survey, *Theoret. Comput. Sci.* 395 (1) (2008) 101–131.
- [24] Y. Mukai, K. Morita, Realizing reversible logic elements with memory in the billiard ball model, *Int. J. Unconv. Comput.* 8 (1) (2012) 47–59.
- [25] F. Peper, J. Lee, S. Adachi, S. Mashiko, Laying out circuits on asynchronous cellular arrays: a step towards feasible nanocomputers? *Nanotechnology* 14 (4) (2003) 469–485.
- [26] L.L. Qian, D. Soloveichik, E. Winfree, Efficient Turing-universal computation with DNA polymers, in: 16th Int. Conf. DNA Computing and Molecular Programming, in: LNCS, vol. 6518, Hong Kong, 2010, 123–140.
- [27] D.P. Vasudevan, P.K. Lala, J. Di, J.P. Parkerson, Reversible-logic design with online testability, *IEEE Trans. Instrum. Meas.* 55 (2) (2006) 406–414.
- [28] R. Wille, An introduction to reversible circuit design, Electronics, Communications and Photonics Conference (SIECP), Saudi International, April 2011.